分布式并行计算环境:MPI

王萃寒 赵 晨 许小刚 吴国新

(东南大学计算机科学与工程系 南京 210096) (教育部计算机网络与信息集成重点实验室 南京 210096)

Distributed Paralel Computing Environment:MPI

WANG Cui-Han ZHAO Chen XU Xiao-Gang WU Guo-Xin

(Department of Computer Science and Engineering,Southeast University,Nanjing 210096)

(Ministry of Educdtion Key Lab of Computer Network and Information Integration,Nanjing 210096)

Abstract Message passing Interface(MPI) is a kind of network distributed parallel computing environments which have been widely used on super parallel computers and networks. First,this paper describes the rssearch background and developing status of MPI. Then on this basis it will study and analyze the functions and features of MPI,summarize its insufficiencies and gives some suggestions for modification.

Keywords Message passing interface,Distributed parallel computing,Point-to-point communication, Collective communication, Communicator,Context

1 引言

在过去几十年里,大规模和超大规模并行机的可用性取得长足进步。由于各种因素,这些机器大多采用分布主存或分布共享主存结构,为了对用户提供必要的支持,厂商开发了各自专有的消息传递包或消息传递库如 Intel 的 NX、IBM 的 EUI、Parasoft 的 Exress、橡树岭的 PVM 等。它们提供了相似的功能,并且在特定平台上具有优越的性能,但是在应用程序一级互不相容,可移植性差。

为了开发一个高效标准具有可移植性的消息传递库,由厂商(如 IBM、Intel 等)、软件开发商(Parasoft、KAI 等)、研究中心(ANL、GMP 等)和大学(Endinburgh、Maryland 等)共同于 1992 年成立了 MPI 论坛。经过共同研究和使用,MPI 论坛先后于 1994 年和 1997 年后提出 MPI-1 和 MPI-2,综合了现有消息传递库的优点,针对并行程序的开发提出了新的概念。自发布以来,得到了众多用户的支持。MPI 使并行计算和处理的应用程序具有了可移植性,且在不同的硬件平台和异质网络上都可获得高性能。

2 MPI 发展现状

定义:MPI 是一种为编写消息传递程序而开发的广泛使用的标准,该接口为消息传递建立了一个实际的、可移植的、有效的和灵活的标准。

MPI-1 具有如下 5 个特点:(1)通用性:MPI 是可移植的标准平台;(2)点对点通信;(3)集体通信方式;(4)MPI 的实现方式多样化;(5)良好的操作环境:具有一定差错控制功能。

1997 年后,MPI-2 又增加了新的特点和内容:(1)并行 I/O;(2)主动消息;(3)动态进程和进程控制。这些新特点使得 MPI 并行库功能更加完善和强大。

虽然 MPI 原本是为 MIMD 大型阵列机的并行计算开发而设计的,但随着当前网络技术的飞速发展,其应用也得到广泛扩展,被应用到网络领域,最典型的就是现在我们经常把 MPI 作为网络中多台计算机并行计算通信环境中的消息传递接口来进行开发,以网络中的计算机作为并行处理和计算的结点,以 MPI 作为并行程序的公共消息传递接口。这样,基于 MPI 开发的并行计算应用程序可以不作任何修改地转移到另一个厂商生产的另一种型号的并行机上运行,或转移到网络环境中运行,只要这些单机(指并行计算机)或网络环境都有或者说都支持 MPI 平台即可。

3 MPI 主要特点

3.1 点对点通信

在点对点通信中,MPI 提供了两类发送和接收机制:阻塞和非阻塞。

· 阻塞发送的完成数据已拷贝出发送缓冲区,即发送缓冲区可以重新分配使用。阻塞接收的完成意味着接收数据已拷贝入接收缓冲区,即接收方已可以使用。但是发送的完成并不意味着一个匹配的接收已发生或完成,发送的消息可能被缓存在系统的缓存区中,这样,发送可在匹配接收发生之前完成。

· 非阻塞操作可以使得计算和通信重叠(必需有系统硬件的支持),MPI 的非阻塞操作立即返回一个句柄,这并不意味着数据已经拷贝出发送缓冲区或拷贝入接收缓冲区,该句柄可用来在适当的时候完成(MPI_Wait)和检测(MPI_Test)对应的非阻塞操作。

基于以上两种通信机制,MPI 提供具有以下语义的阻塞和非阻塞发送:

· 标准方式:由 MPI 根据系统当前的状况选取缓存发送或同步发送方式来完成发送。

· 缓存发送:发送操作可在相应的接收操作发生前发生并完成,其完成则发送缓冲区可重用,但数据的发送其实并没有真正完成,而且把发送数据立即放入系统缓存区中等待相应的接收发生并完成。

· 同步发送:发送操作可在相应的接收操作发生前发生,但是仅在接收完成后才返回(完成),从而实现了发送和接收在通信点上的同步。

· 就绪发送:发送操作仅在相应的接收操作发生后才能发生,其完成后则发送缓冲区可重用。

3.2 集体通信

*)国家自然科学资助项目"下一代网络服务体系结构及其关键技术的研究"。

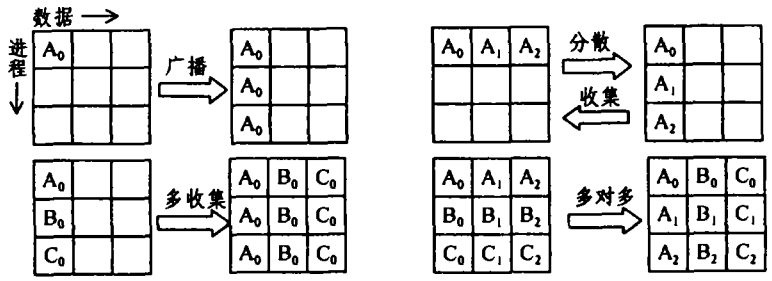

图 1　MPI 集体通信的 5 种组内数据通信操作

所谓集体通信是指涉及一组相关进程的数据处理和通信操作。MPI 提供的集体通信操作函数不但功能强大，而且还能完成对数据的复杂处理。MPI 支持广播(Broadcast)、分散(Scatter)、收集(Gather)、多收集(Allgather)、多对多(All-toall)等进程组内多进程数据通信操作(如图 1 所示)。此外，MPI 还支持进程组内多进程数据运算操作，如 MPI_Reduce()、MPI_AllReduce()等，这使得可以直接在进程组内进行矩阵的加减乘除等复杂运算操作，十分有利于科学计算。

3.3　通信子、组和上下文

· 组(Group)：组定义了一组相关的进程，每个进程有对应的 rank 号，从 0～n－1(n 为组内进程数)，组作为通信子概念的一个组成部分，在通信操作中并不直接使用，只有通信子才在通信操作中直接使用。

· 上下文(Context)：上下文也是通信子的一个组成属性，用于区分消息的通信空间。如在一个组中的两个进程间的通信到底是群体通信还是点对点通信，上下文在 MPI 中是一个隐含对象，用户不可见。

· 通信子(Communicator)：包含进程组、上下文、虚拓扑结构、通信子属性等概念，是 MPI 整个并行计算通信实现的关键技术所在，它为 MPI 中所有的通信操作提供了适合的范围。在 MPI 中，进程以组内的相对的 rank 号来标识，通信子参数则说明所涉及的进程组，使用该通信子的通信操作限制在该进程组的进程之间。这样，一组进程集上的库代码被用于另一组进程集时，库代码无需改动，而只要重定义描述该进程集的通信子。

3.4　进程组虚拓扑结构

MPI 的进程组中每个进程对应的是 rank 号，而在实际中用户设计和处理并行的进程时，并不是直接根据 rank 号，而是根据某种拓扑结构来进行的，通过把拓扑结构映射到 rank 号，使用户能够从通信或处理任务等意义上设计和理解并行处理的进程，而不必要考虑其具体的 rank 号，这种拓扑结构及其映射就是 MPI 的虚拓扑结构。目前，我们采用两种方式来定义和实现进程组的拓扑结构：图和笛卡尔坐标(即用网格方式)。

· 图方式：进程作为结点，以进程间的通信关系作为边，并记录每个结点的度(相邻的结点数)，把这些信息作为通信子的属性保存，从而实现了结点进程和进程 rank 号的对应。

· 笛卡尔坐标方式：每个进程作为一个网格，由一个 n 维的笛卡尔坐标表示，所以我们把整个进程组(假设共有 m 个进程)分成 n 维，每维大小分别为(A1，A2，…，An)，则 m＝A1 * A2 * … * An，从而把进程对应到每个网格。

3.5　动态进程和进程控制

MPI-1 虽然为并行应用程序提供了一个公共消息传递接口，但 MPI-1 的实现是静态的，即所有进程是事先静态分配的，当应用程序启动后，不能从中增加或删除进程结点，这显然不能满足日益增长的网络并行计算模式的需求，即要求能够动态地增加并行处理的进程，动态地增删并行处理的进程和结点，MPI-2 对此在综合了以往一些并行计算接口库的特点的基础上，增加了动态进程和进程控制的机制，允许用户在并行应用程序启动后，动态地生成并行应用进程，并且提供了这些进程和原有进程之间的通信和管理函数。在一个第三方软件的配合下，可以实现 MIMD 环境或网络环境下的资源控制，即动态的增删进程结点，将一个进程从一个结点转移到另一个结点，并允许原结点退出并行计算环境等。

3.6　主动消息

MPI-2 增加了主动消息的新特点。主动消息是一种异步通信机制，其目的在于充分展现硬件的灵活性和互连网的性能。其主要思想是：在每个消息的消息头中包含一个用户级消息处理程序的地址，该程序的入口参数为消息的消息体。通过消息处理程序的执行，将消息从网络中取出，集成到处理节点的当前计算中去。消息处理程序的执行必须快速完成，这与传统的消息传递处理机中，当消息到达时产生中断且执行一段专用的中断处理程序在思想上是统一的。总的来说，主动消息机制只需简单的硬件支持，却可大大降低软件开销，获得近似于消息驱动机的功能特性，对计算和通信的并行有很大好处。

总结　总之，MPI 从整体上说是一个优秀的、标准的分布式并行计算平台，但是 MPI 在某些方面还存在着不足：MPI 没有提供容错的机制，并行应用程序以一个进程组的方式运行时，当进程组中的一个进程或计算机结点失败后，整个并行应用程序都将失败，这虽然保证了当一个并行应用程序失败后，系统中没有残留或挂起进程，但对于失败管理和恢复显然是不足的；在网络并行计算环境中，MPI 也并不直接实现资源管理，换句话说，就是不能在应用程序中增加或删除计算资源(即计算机结点)，或把任务(通常表现为 UNIX 进程)从一个计算机结点转移到另一个计算机结点。

不过，只要通过我们 MPI 的使用者和 MPI 的设计者们的共同努力，还是可以克服 MPI 存在的一些缺点，例如在网络环境中通常我们可以通过实现一个相应的第三方软件来达到增删并行计算结点、转移任务等资源管理功能，从而使 MPI 成为一个理想的、完美的和标准的分布式并行计算平台。

参 考 文 献

1　Message Passing Interface Forum. MPI：A message-passing interface standard. Intl. J. of Supercomputer Applications
2　Nupairoj N，Ni L M. Performance evaluation of some MPI implementaion on workstation clusters. In：Proc. Second Workshop on Scalable Parallel Libraries Conference，Mississippi
3　Gropp W，Lusk E. The Second-Generation ADI for the MPICH Implementation of MPI
4　McBrayn O A. An overview of message passing environment. In Parallel Computing 20，1994
5　Wiel S V，Nathanson D，Lilja D. Performance and program complex in contemporary network-based parallel computing system. University of Minnesota：[Tech Rep：HPPC-96-02]. 1996